Memory

Outline

- Memory Arrays
- SRAM Architecture
  - SRAM Cell
  - Decoders
  - Column Circuitry
  - Multiple Ports
- DRAM
- Serial Access Memories
- ROM
Memory Arrays

Random Access Memory  Serial Access Memory  Content Addressable Memory (CAM)

Read/Write Memory (RAM) (Volatile)  Read Only Memory (ROM) (Nonvolatile)  Shift Registers  Queues

Static RAM (SRAM)  Dynamic RAM (DRAM)  Serial In Parallel Out (SIPO)  Parallel In Serial Out (PISO)

First In First Out (FIFO)  Last In First Out (LIFO)

Mask ROM  Programmable ROM (PROM)  Erasable Programmable ROM (EPROM)  Electrically Erasable Programmable ROM (EEPROM)  Flash ROM

Array Architecture

- $2^n$ words of $2^m$ bits each
- If $n >> m$, fold by $2^k$ into fewer rows of more columns

- Good regularity – easy to design
- Very high density if good cells are used
12T SRAM Cell

- Basic building block: SRAM Cell
  - Holds one bit of information, like a latch
  - Must be read and written
- 12-transistor (12T) SRAM cell
  - Use a simple latch connected to bitline
  - 46 x 75 λ unit cell

6T SRAM Cell

- Cell size accounts for most of array size
  - Reduce cell size at expense of complexity
- 6T SRAM Cell
  - Used in most commercial chips
  - Data stored in cross-coupled inverters
- Read:
  - Precharge bit, bit_b
  - Raise wordline
- Write:
  - Drive data onto bit, bit_b
  - Raise wordline
SRAM Read

- Precharge both bitlines high
- Then turn on wordline
- One of the two bitlines will be pulled down by the cell
- Ex: A = 0, A_b = 1
  - bit discharges, bit_b stays high
  - But A bumps up slightly
- Read stability
  - A must not flip

![SRAM Read Diagram]

SRAM Read

- Precharge both bitlines high
- Then turn on wordline
- One of the two bitlines will be pulled down by the cell
- Ex: A = 0, A_b = 1
  - bit discharges, bit_b stays high
  - But A bumps up slightly
- Read stability
  - A must not flip
  - N1 >> N2
SRAM Write

- Drive one bitline high, the other low
- Then turn on wordline
- Bitlines overpower cell with new value
  - Ex: $A = 0$, $A_b = 1$, $bit = 1$, $bit_b = 0$
    - Force $A_b$ low, then $A$ rises high
- **Writability**
  - Must overpower feedback inverter

---

SRAM Write

- Drive one bitline high, the other low
- Then turn on wordline
- Bitlines overpower cell with new value
  - Ex: $A = 0$, $A_b = 1$, $bit = 1$, $bit_b = 0$
    - Force $A_b$ low, then $A$ rises high
- **Writability**
  - Must overpower feedback inverter
    - $N2 >> P1$
SRAM Sizing

- High bitlines must not overpower inverters during reads
- But low bitlines must write new value into cell

![SRAM Sizing Diagram](image)

SRAM Column Example

**Read**

Read operation involves bitline conditioning and activation of word and bit lines.

**Write**

Write operation involves writing the new data to the cell. The data is sent through a multiplexer controlled by the write enable signal.
SRAM Layout

- Cell size is critical: 26 x 45 λ
- Tile cells sharing V_{DD}, GND, bitline contacts

Decoders

- \( n:2^n \) decoder consists of \( 2^n \) n-input AND gates
  - One needed for each row of memory
  - Build AND from NAND or NOR gates

Static CMOS

Pseudo-nMOS
Decoder Layout

- Decoders must be pitch-matched to SRAM cell
  - Requires very skinny gates

![Decoder Layout Diagram]

Large Decoders

- For n > 4, NAND gates become slow
  - Break large gates into multiple smaller gates
Predecoding

- Many of these gates are redundant
  - Factor out common gates into predecoder
  - Saves area
  - Same path effort

Column Circuitry

- Some circuitry is required for each column
  - Bitline conditioning
  - Sense amplifiers
  - Column multiplexing
Bitline Conditioning

- Precharge bitlines high before reads
  - ![Diagram of precharge](image)
  - Equalize bitlines to minimize voltage difference when using sense amplifiers
    - ![Diagram of equalization](image)

Sense Amplifiers

- Bitlines have many cells attached
  - Ex: 32-kbit SRAM has 256 rows x 128 cols
  - 128 cells on each bitline

- \( t_{pd} \propto (C/I) \Delta V \)
  - Even with shared diffusion contacts, 64C of diffusion capacitance (big C)
  - Discharged slowly through small transistors (small I)

- Sense amplifiers are triggered on small voltage swing (reduce \( \Delta V \))
**Twisted Bitlines**

- Sense amplifiers also amplify noise
  - Coupling noise is severe in modern processes
  - Try to couple equally onto bit and bit_b
  - Done by *twisting* bitlines

![Diagram of twisted bitlines]

**Column Multiplexing**

- Recall that array may be folded for good aspect ratio
- Ex: 2 kword x 16 folded into 256 rows x 128 columns
  - Must select 16 output bits from the 128 columns
  - Requires 16 8:1 column multiplexers
Tree Decoder Mux

- Column mux can use pass transistors
  - Use nMOS only, precharge outputs
- One design is to use $k$ series transistors for $2^k$:1 mux
  - No external decoder logic needed

Single Pass-Gate Mux

- Or eliminate series transistors with separate decoder
Ex: 2-way Muxed SRAM

Multiple Ports

- We have considered single-ported SRAM
  - One read or one write on each cycle

- Multiported SRAM are needed for register files

- Examples:
  - Multicycle MIPS must read two sources or write a result on some cycles
  - Pipelined MIPS must read two sources and write a third result each cycle
  - Superscalar MIPS must read and write many sources and results each cycle
**Dual-Ported SRAM**

- Simple dual-ported SRAM
  - Two independent single-ended reads
  - Or one differential write

- Do two reads and one write by time multiplexing
  - Read during ph1, write during ph2

**Multi-Ported SRAM**

- Adding more access transistors hurts read stability
- Multiported SRAM isolates reads from state node
- Single-ended design minimizes number of bitlines
Dynamic RAM (DRAM)

- Capacitor can hold charge
- Transistor acts as gate
- No charge is a 0
- Can add charge to store a 1
- Then open switch (disconnect)
- Can read by closing switch

Precharge and Sense Amps

- You’ll see “precharge time”
- B is precharged to $\frac{1}{2} V$
- Charge/no-charge on C will increase or decrease voltage
- Sense amps detect this
DRAM Characteristics

- **Destructive Read**
  - When cell read, charge removed
  - Must be restored after a read

- **Refresh**
  - Also, there’s steady leakage
  - Charge must be restored periodically

Serial Access Memories

- Serial access memories do not use an address
  - Shift Registers
  - Tapped Delay Lines
  - Serial In Parallel Out (SIPO)
  - Parallel In Serial Out (PISO)
  - Queues (FIFO, LIFO)
Shift Register

- Shift registers store and delay data
- Simple design: cascade of registers
  - Watch your hold times!

Denser Shift Registers

- Flip-flops aren’t very area-efficient
- For large shift registers, keep data in SRAM instead
- Move read/write pointers to RAM rather than data
  - Initialize read address to first entry, write to last
  - Increment address on each cycle
Tapped Delay Line

- A tapped delay line is a shift register with a programmable number of stages
- Set number of stages with delay controls to mux
  - Ex: 0 – 63 stages of delay

Serial In Parallel Out

- 1-bit shift register reads in serial data
  - After N steps, presents N-bit parallel output
Parallel In Serial Out

- Load all N bits in parallel when shift = 0
- Then shift one bit out per cycle

Queues

- Queues allow data to be read and written at different rates.
- Read and write each use their own clock, data
- Queue indicates whether it is full or empty
- Build with SRAM and read/write counters (pointers)
FIFO, LIFO Queues

- **First In First Out (FIFO)**
  - Initialize read and write pointers to first element
  - Queue is EMPTY
  - On write, increment write pointer
  - If write almost catches read, Queue is FULL
  - On read, increment read pointer

- **Last In First Out (LIFO)**
  - Also called a stack
  - Use a single stack pointer for read and write

Read-Only Memories

- Read-Only Memories are nonvolatile
  - Retain their contents when power is removed

- Mask-programmed ROMs use one transistor per bit
  - Presence or absence determines 1 or 0
**ROM Example**

- **4-word x 6-bit ROM**
  - Represented with dot diagram
  - Dots indicate 1’s in ROM

Word 0: 010101
Word 1: 011001
Word 2: 100101
Word 3: 101010

Looks like 6 4-input pseudo-nMOS NORs

**ROM Array Layout**

- **Unit cell is 12 x 8 (about 1/10 size of SRAM)**
PROMs and EPROMs

- **Programmable ROMs**
  - Build array with transistors at every site
  - Burn out fuses to disable unwanted transistors

- **Electrically Programmable ROMs**
  - Use floating gate to turn off unwanted transistors
    - Trapped negative charges on the floating gate increase $V_t$ and shut off the transistor all the time
  - EPROM, EEPROM, Flash

- **Erasable Programmable ROM – EPROM**
  - Programmed electronically
  - Erased through exposure to UV light

- **Electrically Erasable Programmable ROM – EEPROM**
  - Electronically programmable/erasable in bytes

- **Flash**
  - Electrically Programmable in bytes
  - Electrically erasable in large blocks or entire chip (early flash)
  - Can be thought as a special EEPROM
  - NAND flash (vs. NOR flash): high density, less expensive; mainstream for today’s memory cards & USB flash drives.